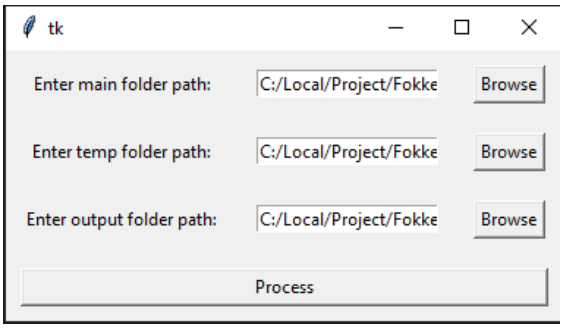
GETTING STARTED

➢ Necessary libraries to be installed (requirement.txt)

1) fitz: This library is used for working with PDF files. You can install it using the command: **pip install PyMuPDF**
2) PIL (Python Imaging Library): This library is used for image processing and manipulation. You can install it using the command: **pip install Pillow**
3) cv2 (OpenCV): This library is used for computer vision tasks, including image processing and analysis. You can install it using the command**:**
   pip install opencv-python.
4) numpy: This library is used for numerical computations and array manipulation. You can install it using the command: **pip install numpy**
5) easyocr: This library is used for optical character recognition (OCR) to extract text from images. You can install it using the command: **pip install easyocr**
6) pandas: This library is used for data manipulation and analysis. You can install it using the command: **pip install pandas**

➢ How to run the code

1) Once the libraries are installed, save and run the code.
2) It will ask for three inputs from the user which are as follows:



   a. Enter the path of the parent input folder (which contains the pdf files): #Takes the main parent folder as the input

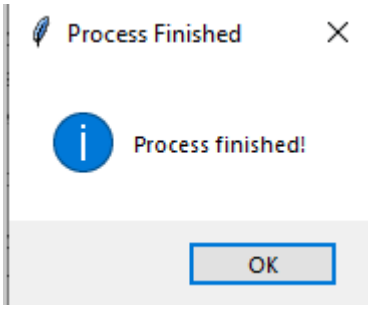
b. Enter the path of the intermediate output folder (chunks and images will be stored): #Intermediate output folder which stores all the Images of the pdf files. This will be deleted at the end

c. Enter the path of the final output folder where the excel files will be stored: #Final output folder where the excel files corresponding to each pdf file will stored

3) It will start executing.

```
(project) PS C:\Local\Project> & c:/tools/pyenvs/project/Scripts/python.exe c:/Local/Project/sample_gui.py
Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is much faster with a GPU.
input:C:/Local/Project/Fokker/Sample_Input_Output/SampleInput
preprocess:C:/Local/Project/Fokker/Sample_Input_Output/bbb
output:C:/Local/Project/Fokker/Sample_Input_Output/aaa
Processing Data:  50%|
```

4) Once the execution is done a pop up window will be displayed.




```
Output saved to C:/Local/Project/Fokker/Sample_Input_Output/aaa\286A5111-2359-WRR0066\286A5111-2359-WRR0066 (V5)\286A5111-2359-WRR0066.xlsx successfully.
Processing Data: 100%|                                                                                  | 100/100 [06:46<00:00,  4.07s/%]
(project) PS C:\Local\Project>
```

5) Click on Ok. The output will be found in the output folder given by you.

➢ Code Explanation

This code consists of multiple functions and operations to perform the following tasks:

1. Convert PDF to Images

The function convert_pdf_to_images takes the main parent folder (input folder) as input and walks through all the subfolders in it. It converts each PDF file to an image and stores it in the output folder. The function uses the following libraries: fitz, PIL, os, shutil, cv2, numpy, easyocr, re, and pandas.

2. Chunk the Images

The function has_data is used to check if a chunk of an image has any significant information. The crop_image_into_chunks function crops the images into chunks for readability purposes. It takes an image path as input, opens the image, and crops it into chunks based on the provided chunk width and height. The function uses the Image and cv2 libraries.

3. Generate Excel Files

The function process_image reads each image and extracts colors and numbers from the text using the easyocr library. The process_subfolder function processes the images in a subfolder by calling the process_image function. It uses multithreading to process multiple chunks simultaneously. The process_folder function replicates the input directory structure and calls the process_subfolder function for each subfolder.

The extracted colors and numbers are stored in a Pandas DataFrame, and the DataFrame is saved as an Excel file in the specified output folder. The function uses the easyocr, os, re, and pandas libraries.

Finally, the intermediate folder containing all the chunks and images is removed using the shutil library.